

LAPORAN HASIL KELOMPOK
PENJUALAN TIKET BUS



DOSEN PENGAMPU :

Agung Nugroho, S.Kom, M.Kom.

DISUSUN OLEH :

ALFAZA PUTRA ADJIE ARIEFANSYAH 312210512

FEIBERT SIANTURI 312210578

MUHAMMAD HASBIALLAH 312210618

RIDHA MUHAMMAD RIFQI 312210491

ROBBY FIRMANSYAH 312210643

TI.22.A.5

UNIVERSITAS PELITA BANGSA

FAKULTAS TEKNIK

2022/2023

BAB I

PENDAHULUAN

1.1. Latar Belakang

Latar belakang dari pembuatan praktikum join tabel di atas adalah untuk mempelajari dan memahami konsep join dalam pemrograman basis data. Join merupakan salah satu operasi yang penting dalam SQL yang memungkinkan penggabungan data dari beberapa tabel berdasarkan kolom yang memiliki nilai yang sama.

Dalam pengembangan aplikasi atau sistem manajemen basis data, seringkali kita perlu mengambil data yang terkait dari beberapa tabel yang berbeda. Misalnya, kita ingin menampilkan data peserta kuliah beserta nama dosen pengajar dan nama matakuliah yang diambil. Dalam situasi seperti itu, join tabel digunakan untuk menggabungkan data dari tabel-tabel yang relevan berdasarkan hubungan antara kolom-kolom yang terkait.

Pembuatan praktikum join tabel bertujuan untuk memberikan pemahaman dan keterampilan dalam melakukan join antara tabel-tabel yang berbeda, mengerti konsep foreign key dan primary key untuk menghubungkan tabel-tabel tersebut, serta mengambil data yang terkait melalui join. Dengan menguasai konsep dan penggunaan join, akan memudahkan pengembangan aplikasi yang melibatkan pemrosesan data dari beberapa tabel yang terkait.

Pada laporan praktikum, akan dijelaskan langkah-langkah praktis untuk melakukan join tabel, contoh-contoh penggunaan join, serta manfaat dan pentingnya pemahaman mengenai join dalam pemrograman basis data. Laporan praktikum ini bertujuan untuk memberikan pemahaman yang mendalam tentang konsep join dan kemampuan praktis dalam menggunakannya dalam pengembangan aplikasi basis data.

1.2. Tujuan penulisan

Tujuan dari praktikum ini adalah:

1. Mahasiswa mengenal struktur mySQL
2. Mahasiswa mengenal perintah-perintah atau syntax mySQL

BAB II

DASAR TEORI

2.1. Konsep Penggunaan mySQL

Dasar teori dalam pembahasan join tabel dalam pemrograman basis data meliputi konsep relasi, kunci utama (primary key), kunci asing (foreign key), dan operasi join.

1. **Konsep Relasi:**
Relasi adalah representasi struktur data dalam bentuk tabel yang terdiri dari baris (record) dan kolom (field). Setiap tabel memiliki nama yang unik dan terdiri dari satu atau lebih kolom yang memiliki tipe data yang berbeda. Relasi merupakan dasar dari model data relasional dalam basis data.
2. **Kunci Utama (Primary Key):**
Kunci utama adalah kolom atau kombinasi kolom yang unik dan secara unik mengidentifikasi setiap baris dalam tabel. Setiap tabel harus memiliki kunci utama yang membedakan setiap baris secara unik.
3. **Kunci Asing (Foreign Key):**
Kunci asing adalah kolom atau kombinasi kolom dalam sebuah tabel yang mengacu pada kunci utama (primary key) di tabel lain. Kunci asing digunakan untuk menghubungkan dua tabel yang memiliki relasi atau hubungan antara mereka.
4. **Operasi Join:**
Operasi join adalah operasi yang digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan kolom yang memiliki nilai yang sama. Terdapat beberapa jenis operasi join, antara lain:

Inner Join: Menggabungkan baris yang memiliki nilai yang sama pada kolom yang dijadikan acuan.

Left Join: Menggabungkan semua baris dari tabel kiri dan baris yang memiliki nilai yang sama pada kolom yang dijadikan acuan dari tabel kanan.

Right Join: Menggabungkan semua baris dari tabel kanan dan baris yang memiliki nilai yang sama pada kolom yang dijadikan acuan dari tabel kiri.

Full Outer Join: Menggabungkan semua baris dari kedua tabel, termasuk baris yang tidak memiliki nilai yang sama pada kolom yang dijadikan acuan.

Operasi join memungkinkan pengguna untuk mengambil data yang terkait dari beberapa tabel, memperluas kemampuan kueri dan analisis data dalam basis data relasional. Dengan menggunakan konsep kunci utama dan kunci asing, kita dapat menghubungkan tabel-tabel secara terstruktur dan mengambil informasi yang lebih lengkap dan terintegrasi melalui operasi join.

BAB III

3.1. ISI LAPORAN PRAKTIKUM

Dengan menggunakan program mySQL Workbench dan membuat database/schema baru.

Langkah – Langkah sebagai berikut.

- **SCRIPT Penjualan Tiket Bus**

```
CREATE DATABASE penjualan_tiket_bus;  
USE penjualan_tiket_bus;
```

-- Tabel Armada

```
CREATE TABLE data_armada (kode_bus INT PRIMARY KEY,  
nama_bus VARCHAR(25), alamat_tujuan VARCHAR(25), keberangkatan VARCHAR(30));
```

```
INSERT INTO data_armada (kode_bus, nama_bus, alamat_tujuan, keberangkatan)  
VALUES
```

```
    ('1012', 'Murni Jaya', 'Palembang', 'Bekasi'),  
    ('1013', 'Sinar Abadi', 'Madura', 'Jakarta'),  
    ('1014', 'Maju Lancar', 'Semarang', 'Cikarang'),  
    ('1015', 'Gunung Harapan', 'Yogyakarta', 'Sukaresmi');
```

-- Tabel Jadwal Keberangkatan

```
CREATE TABLE jadwal_keberangkatan  
(kode_keberangkatan INT PRIMARY KEY, tgl_keberangkatan VARCHAR(30), kode_bus INT);
```

```
INSERT INTO jadwal_keberangkatan (kode_keberangkatan, tgl_keberangkatan, kode_bus)  
VALUES
```

```
    ('66184', '2024-07-25', '1012'),  
    ('66187', '2024-06-27', '1013'),  
    ('66189', '2024-07-13', '1014'),  
    ('66190', '2024-06-28', '1015');
```

```
ALTER TABLE jadwal_keberangkatan ADD FOREIGN KEY (kode_bus) REFERENCES data_armada  
(kode_bus);
```

-- Tabel Data penumpang dan Posisi Tempat Duduk

```
CREATE TABLE Data_Penumpang_dan_Posisi_Tempat_Duduk  
(kode_penumpang INT PRIMARY KEY,  
nomor_tempat_duduk VARCHAR(25),  
nama VARCHAR(20),  
kode_bus INT);
```

-- Input database nya dulu sebelum dijadikan Foreign Key

```
INSERT INTO Data_Penumpang_dan_Posisi_Tempat_Duduk (kode_penumpang,  
nomor_tempat_duduk, nama, kode_bus)
```

VALUES

```
('2812', '11-20', 'Ahmad', '1012'),  
( '2813', '21-30', 'Rio', '1013'),  
( '2814', '31-40', 'Rina', '1014'),  
( '2815', '40-45', 'Rani', '1015');
```

-- Baru Penambahan Foregin Key pada kolom kode_bus di data_penumpang_dan_posisi_tempat_duduk

```
ALTER TABLE data_penumpang_dan_posisi_tempat_duduk ADD FOREIGN KEY (kode_bus)  
REFERENCES data_armada (kode_bus);
```

-- Tabel Transaksi Penjualan

```
CREATE TABLE Transaksi_Penjualan (kode_transaksi_penjualan INT PRIMARY KEY,  
kode_keberangkatan INT,  
kode_penumpang INT,  
tgl_transaksi_penjualan DATE,  
nomor_tempat_duduk INT);
```

-- Input database nya dulu sebelum dijadikan Foregin Key

```
INSERT INTO Transaksi_Penjualan (kode_transaksi_penjualan, kode_keberangkatan,  
kode_penumpang, tgl_transaksi_penjualan, nomor_tempat_duduk)  
VALUES
```

```
('101513', '66189', '2813', '2024-07-08', '022'),  
( '101414', '66190', '2815', '2024-06-18', '042'),  
( '101215', '66184', '2812', '2024-07-14', '014'),  
( '101316', '66187', '2814', '2024-06-20', '036');
```

-- Baru Penambahan Foregin Key pada Kolom Kode penumpang di table Transaksi Penjualan

```
ALTER TABLE transaksi_penjualan ADD FOREIGN KEY (kode_penumpang)  
REFERENCES Data_penumpang_dan_posisi_tempat_duduk (kode_penumpang);
```

-- Tabel Laporan Transaksi

```
CREATE TABLE Laporan_Transaksi (kode_transaksi INT PRIMARY KEY,  
kode_transaksi_penjualan INT,  
tgl_transaksi_penjualan DATE,  
jumlah_pembayaran VARCHAR(50),  
FOREIGN KEY (kode_transaksi_penjualan) REFERENCES  
Transaksi_penjualan(kode_transaksi_penjualan));
```

-- Input database nya dulu sebelum dijadikan Foregin Key

-- Menginput Data Laporan Transaksi

```
INSERT INTO laporan_transaksi (kode_transaksi, kode_transaksi_penjualan,  
tgl_transaksi_penjualan, jumlah_pembayaran)  
VALUES
```

```
('2021513', '101513', '2024-07-08', '300.000'),  
( '2031414', '101414', '2024-06-18', '500.000'),  
( '2041215', '101215', '2024-07-14', '170.000'),  
( '2051316', '101316', '2024-06-20', '700.000');
```

-- Baru Penambahan Foreign Key pada Kolom Kode penumpang di table Laporan_transaksi

```
ALTER TABLE laporan_transaksi ADD FOREIGN KEY (kode_transaksi_penjualan)
REFERENCES transaksi_penjualan (kode_transaksi_penjualan);
```

```
UPDATE jadwal_keberangkatan set tgl_keberangkatan = "2024-06-27" WHERE kode_keberangkatan =
"66187";
```

```
DELETE from jadwal_keberangkatan WHERE kode_keberangkatan;
```

- **SQL JOIN :**

1. Select *from transaksi_penjualan
Join data_penumpang_dan_posisi_tempat_duduk on
transaksi_penjualan.kode_penumpang=data_penumpang_dan_posisi_tempat_duduk.kode_penumpang;

Output :

	kode_transaksi_penjualan	kode_keberangkatan	kode_penumpang	tgl_transaksi_penjualan	nomor_tempat_duduk	kode_penumpang	nomor_tempat_duduk	nama	kode_bus
►	101215	66184	2812	2024-07-14	14	2812	11-20	Ahmad	1012
	101316	66187	2814	2024-06-20	36	2814	31-40	Rina	1014
	101414	66190	2815	2024-06-18	42	2815	40-45	Rani	1015
	101513	66189	2813	2024-07-08	22	2813	21-30	Rio	1013

2. Select *from data_armada
join data_penumpang_dan_posisi_tempat_duduk on
data_armada.kode_bus=data_penumpang_dan_posisi_tempat_duduk.kode_bus;

Output :

	kode_bus	nama_bus	alamat_tujuan	keberangkatan	kode_penumpang	nomor_tempat_duduk	nama	kode_bus
►	1012	Murni Jaya	Palembang	Bekasi	2812	11-20	Ahmad	1012
	1013	Sinar Abadi	Madura	Jakarta	2813	21-30	Rio	1013
	1014	Maju Lancar	Semarang	Cikarang	2814	31-40	Rina	1014
	1015	Gunung Harapan	Yogyakarta	Sukaresmi	2815	40-45	Rani	1015

3. Select data_armada.kode_bus, data_armada.nama_bus from data_armada
join data_penumpang_dan_posisi_tempat_duduk on
data_armada.kode_bus=data_penumpang_dan_posisi_tempat_duduk.kode_bus;

Output :

	kode_bus	nama_bus
►	1012	Murni Jaya
	1013	Sinar Abadi
	1014	Maju Lancar
	1015	Gunung Harapan

4. Select kode_keberangkatan,kode_penumpang,tgl_transaksi_penjualan,kode_bus
From data_penumpang_dan_posisi_tempat_duduk inner join transaksi_penjualan using
(kode_penumpang);

Output :

	kode_keberangkatan	kode_penumpang	tgl_transaksi_penjualan	kode_bus
▶	66184	2812	2024-07-14	1012
	66189	2813	2024-07-08	1013
	66187	2814	2024-06-20	1014
	66190	2815	2024-06-18	1015

5. Select data_armada.
kode_bus,nama_bus,alamat_tujuan,keberangkatan,data_penumpang_dan_posisi_tempat_duduk.kode_penumpang,nomor_tempat_duduk from data_armada left join
data_penumpang_dan_posisi_tempat_duduk using (kode_bus);

Output :

	kode_bus	nama_bus	alamat_tujuan	keberangkatan	kode_penumpang	nomor_tempat_duduk
▶	1012	Murni Jaya	Palembang	Bekasi	2812	11-20
	1013	Sinar Abadi	Madura	Jakarta	2813	21-30
	1014	Maju Lancar	Semarang	Cikarang	2814	31-40
	1015	Gunung Harapan	Yogyakarta	Sukaresmi	2815	40-45

6. select *from data_armada right join jadwal_keberangkatan using (kode_bus);

Output :

	kode_bus	kode_keberangkatan	tgl_keberangkatan	nama_bus	alamat_tujuan	keberangkatan
▶	1012	66184	2024-07-25	Murni Jaya	Palembang	Bekasi
	1013	66187	2024-06-27	Sinar Abadi	Madura	Jakarta
	1014	66189	2024-07-13	Maju Lancar	Semarang	Cikarang
	1015	66190	2024-06-28	Gunung Harapan	Yogyakarta	Sukaresmi

7. select *from data_armada,data_penumpang_dan_posisi_tempat_duduk where
data_armada.kode_bus=data_penumpang_dan_posisi_tempat_duduk.kode_bus;

Output :

	kode_bus	nama_bus	alamat_tujuan	keberangkatan	kode_penumpang	nomor_tempat_duduk	nama	kode_bus
▶	1012	Murni Jaya	Palembang	Bekasi	2812	11-20	Ahmad	1012
	1013	Sinar Abadi	Madura	Jakarta	2813	21-30	Rio	1013
	1014	Maju Lancar	Semarang	Cikarang	2814	31-40	Rina	1014
	1015	Gunung Harapan	Yogyakarta	Sukaresmi	2815	40-45	Rani	1015

8. select *from laporan_transaksi join transaksi_penjualan on
laporan_transaksi.kode_transaksi_penjualan=transaksi_penjualan.kode_transaksi_penjualan;

Output :

	kode_transaksi	kode_transaksi_penjualan	tgl_transaksi_penjualan	jumlah_pembayaran	kode_transaksi_penjualan	kode_keberangkatan	kode_penumpang	tgl_transaksi_penjualan	nomor_tempat_du
▶	2021513	101513	2024-07-08	300.000	101513	66189	2813	2024-07-08	22
	2031414	101414	2024-06-18	500.000	101414	66190	2815	2024-06-18	42
	2041215	101215	2024-07-14	170.000	101215	66184	2812	2024-07-14	14
	2051316	101316	2024-06-20	700.000	101316	66187	2814	2024-06-20	36

- **OUTPUT SCRIPT :**

1. Data Armada

	kode_bus	nama_bus	alamat_tujuan	keberangkatan
▶	1012	Murni Jaya	Palembang	Bekasi
	1013	Sinar Abadi	Madura	Jakarta
	1014	Maju Lancar	Semarang	Cikarang
	1015	Gunung Harapan	Yogyakarta	Sukaresmi
*	NULL	NULL	NULL	NULL

2. Jadwal Keberangkatan

	kode_keberangkatan	tgl_keberangkatan	kode_bus
▶	66184	2024-07-25	1012
	66187	2024-06-27	1013
	66189	2024-07-13	1014
	66190	2024-06-28	1015
*	NULL	NULL	NULL

3. Data penumpang dan Posisi Tempat Duduk

	kode_penumpang	nomor_tempat_duduk	nama	kode_bus
▶	2812	11-20	Ahmad	1012
	2813	21-30	Rio	1013
	2814	31-40	Rina	1014
	2815	40-45	Rani	1015
*	NULL	NULL	NULL	NULL

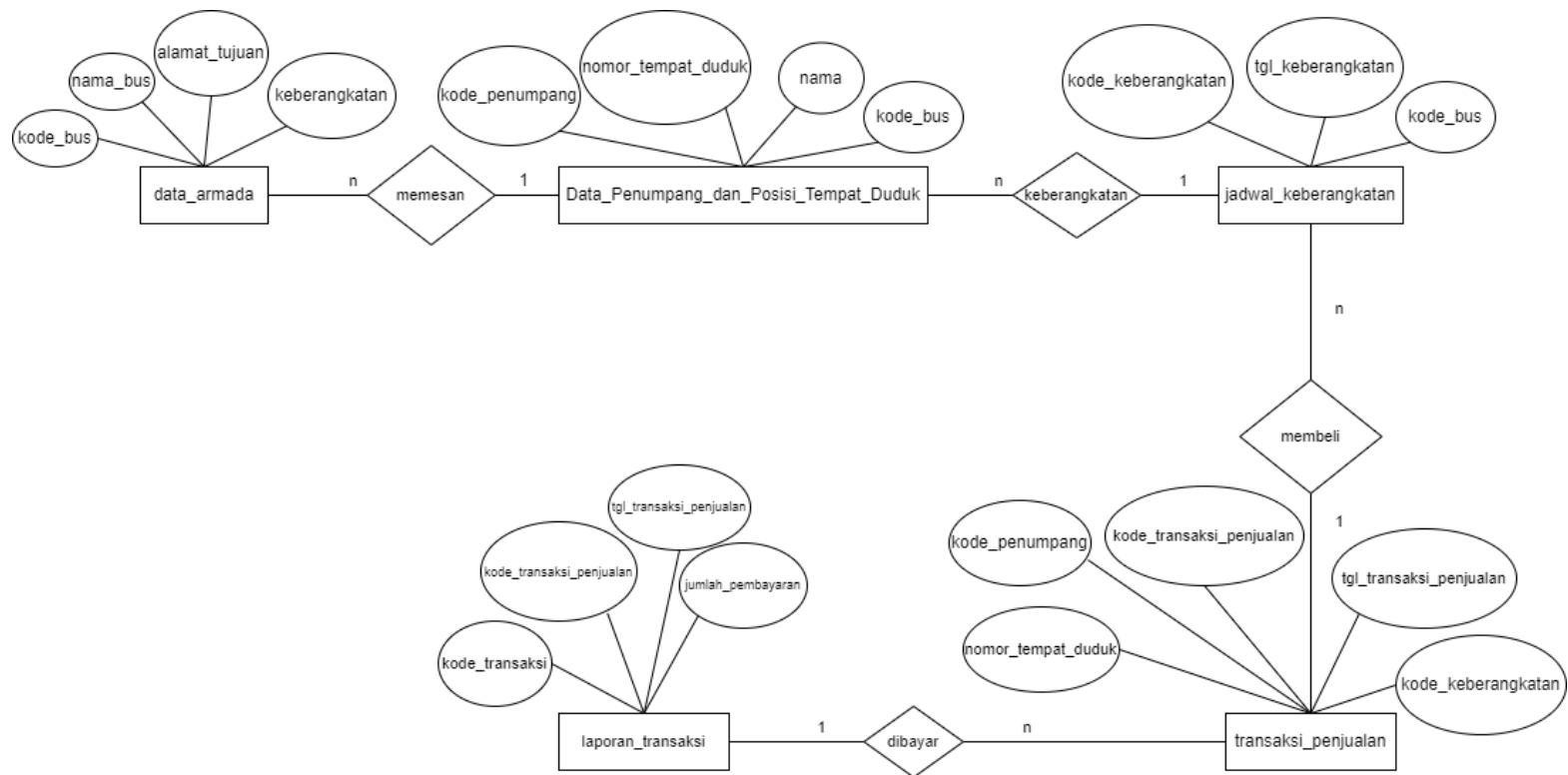
4. Transaksi Penjualan

	kode_transaksi_penjualan	kode_keberangkatan	kode_penumpang	tgl_transaksi_penjualan	nomor_tempat_duduk
▶	101215	66184	2812	2024-07-14	14
	101316	66187	2814	2024-06-20	36
	101414	66190	2815	2024-06-18	42
	101513	66189	2813	2024-07-08	22
*	NULL	NULL	NULL	NULL	NULL

5. Laporan Transaksi

	kode_transaksi	kode_transaksi_penjualan	tgl_transaksi_penjualan	jumlah_pembayaran
▶	2021513	101513	2024-07-08	300.000
	2031414	101414	2024-06-18	500.000
	2041215	101215	2024-07-14	170.000
	2051316	101316	2024-06-20	700.000
*	NULL	NULL	NULL	NULL

- **ERD Penjualan Tiket Bus**



BAB IV

PENUTUP

3.1. KESIMPULAN

Pembuatan database penjualan tiket bus bertujuan untuk memberikan solusi yang terorganisir dalam mengelola data penumpang, memantau data armada, dan melacak transaksi

pembayaran yang terkait. Dengan menggunakan database, informasi armada dapat diakses dengan mudah dan cepat, meminimalkan risiko kesalahan dalam proses administrasi, serta mempercepat proses pelayanan di armada-armada yang berbeda.

Untuk membuat tabel, kita bisa menggunakan bahasa SQL dan menuliskan secara manual sesuai dengan sintaks bahasa SQL atau kita bisa membuat tabel dengan langsung menggunakan GUI yang tersedia di server MySQL.

3.2. SARAN

Ada pun saran yang dapat kami sampaikan yaitu sebaiknya modul yang telah diberikan lebih baik di perbaharui sesuai dengan software terbaru, agar mahasiswa tidak kebingungan saat mengerjakan Tugas ataupun Praktikum.